

# Suspended Traffic Lights Detection and Distance Estimation Using Color Features

Moises Diaz-Cabrera<sup>1</sup>, Pietro Cerri<sup>2</sup> and Javier Sanchez-Medina<sup>3</sup>, *Member, IEEE*

**Abstract**—Traffic Light Detection is a problem differently approached by many research groups around the world. Here we present a novel technique to detect suspended traffic lights, based on colors and features such as black area of traffic lights or area of lighting lamps. Additionally, the traffic light distance is estimated aiming at slowing down and stopping in the correct position, in case of red light. Some preliminary test results are presented to assess both the detection rate and the distance estimation.

## I. INTRODUCTION

Research on traffic light detection is widely present in literature. The presence of traffic lights in urban environments cannot be ignored by research groups that develop advanced driver assistance systems or work on the topic of autonomous driving. Radio Frequency IDentification (RFID), and other Vehicle-to-Infrastructure (V2I) communication technologies could potentially substitute traffic lights in an only autonomous vehicles scenario. However, in a mixed manual-autonomous vehicle situation – more likely to happen in a near future – traffic control will still need traffic lights, specially in urban traffic networks.

We have extended our target to not just detect traffic light stages, but also to estimate their distance. A real scene studied is shown in figure 1. In order to achieve this goal, we take advantage from traffic light features, camera calibration, and image processing. A simple tracking stage is added to reach more robust results and to cope with blinking noise. The developed algorithm is based on simple rules and simple processing in order to guarantee a safe real time execution.

In the next section the state of the art in traffic light detection is presented. In section II the algorithm is presented, in section III results are presented. Finally, in section IV some concluding remarks and our future development plans are shared.

### A. State of art

Many computer vision groups are working on traffic light detection: a wide variety of approaches is present in the literature. An easy way to detect traffic lights is shown

<sup>1</sup>Moises Diaz-Cabrera (corresponding author) is PhD candidate at CICEI (www.cicei.com), ULPGC moisdca@gmail.com

<sup>2</sup>Pietro Cerri is research engineer at VisLab (www.vislab.com) and high skill technician at University of Parma, Italy cerri@vislab.com

<sup>3</sup>J. Sanchez-Medina is researcher at CICEI and Assistant Professor at DIS (Computer Science Department), both belonging to the University of Las Palmas de Gran Canaria (ULPGC), Spain javier.sanchez.medina@ieee.org



Fig. 1. Real scene with traffic lights on the road

in [1]. The authors present a system based on traffic light features, such as color, area and density of pixels. Their proposed methodology seems successful for a fixed scenario in day-time, but not at night-time. Furthermore, this approach reaches good results also in arrow detection with simple but successful rules.

In [2] a low cost camera combined with a robust and efficient algorithm is used. From a Gaussian distribution, they model the hue and saturation parameters for a set of training traffic light images, in order to learn the parameters. Then they look for traffic light regions in test images using the modelled parameters.

A large set of traffic light images is used in [3]. Images contain several traffic light positions and different states. A 2040×1080px region-of-interest (ROI) is used in each image. Indeed, since the camera lens has a 30° field of view, they propose a detection algorithm to detect traffic lights from up to 150 m distances.

Also, they resolve the brightness problem from the LED lights, by adjusting the shutter. According to this problem, we have reduced the shutter speed, setting it between 1/100 and 1/200 s<sup>-1</sup>. The methodology developed by the authors is divided into four parts: prediction of the distance between car and traffic lights, classification to fix the bounding box, selection of the correct class of traffic light, and finally, confidence degradation by filtering in unobserved lights over time. Additionally, they confirm that watching for a relative change in intensity of the light elements could be necessary in the case of arrows, because their light correction algorithm is very weak.

Another similar research is shared in [4], where their target is on-line traffic lights detection in urban and rural environments. They have developed several traffic light

recognition systems based on learning processes, such as cascade classifiers with AdaBoost. Their system was validated in real conditions using registered traffic light video recording from various countries. Processing was done in real-time on 640x480 images using a 2.9GHz single core desktop computer. Their method reached up to 95% of precision which is better than the results achieved with cascade classifiers. Three steps are taken: a robust spot light detection applied to grayscale images; then, an adaptive template matcher looks for matching confidence; finally a validation step is executed.

Finally, other approaches take advantage of the Hough transform ([5] and [6]). This new method considers the structure of traffic lights based on the Hough transform. They have defined a model of traffic light for the detection. The camera was situated behind the windshield during the test. In the proposed method, given an input image, each pixel of the image is clustered. Our method's clusterization (see subsection II-A) is partially inspired by this paper. The pixels with the colors of the traffic light are only used for edge detection. Finally, a traffic light is detected by voting with the model and detected edge.

Briefly, our method is based mainly on detection in color frames. Subsequently, we search for colors and visual characteristics similar to the features of traffic lights. Then, a tracking stage helps to keep those areas that are proper traffic lights and to filter the ones that are not. Thus we get a method that works in real time from a set of simple rules.

## II. DESCRIPTION OF THE METHOD

Camera parameters, both intrinsic and extrinsic, are known, the maximum and minimum height of traffic lights can be estimated according to Italian traffic laws, as well. Therefore, in order to speed up the processing, it is possible to compute the lower row in which a traffic light could be found. Considering this value it is possible to calculate a Region of Interest that will be processed to search for traffic lights.

### A. Color based clustering

Color detection and clusterization is a well known problem in computer vision. Traffic light image RGB components may be affected by different variations in scene illumination, optics drop-off, variable distance, etc. Having this in mind, we have extracted the RGB components of 60 different raw images and from their respective 60 RGB normalized images too. They were specifically selected to cover very varied conditions like fog, snow, clear day or night time illumination. Then, we manually selected the red, amber and green pixels in each image. For each pixel we have six values: the three RGB components ( $R, G, B$ ), and the three normalized RGB components ( $R_N, G_N, B_N$ ). All components were then combined in order to get sharp borders on the red, amber and green colors of the pictures captured by our camera.

After considering a wide range of possibilities, it was possible to find the best and clear borders by using the values represented in the box-and-whisker plot in figure 2.

Therefore, when a pixel satisfies one rule, it is categorized into a cluster. We have set up five clusters attending to the following rules:

Cluster 1 (*red color*):  $R_N - G_N > \alpha$  and  $G_N - B_N < \beta$  and  $G_N < \gamma$

Cluster 2 (*amber color*):  $R - B > \delta$  and  $G_N - B_N > \omega$  and  $G_N > \pi$

Cluster 3 (*green color*):  $R_N - G_N < \mu$  and  $G_N - B_N > \rho$  and  $G_N > \epsilon$

Cluster 4 (*black color*):  $R + G + B < \varphi$

Cluster 5 (*white color*): *Otherwise*

The conversion carried out is with L0 - norm:

$$R_N, G_N, B_N = \begin{cases} 0, & \text{for } s = 0 \\ \frac{R}{s}, \frac{G}{s}, \frac{B}{s}, & \text{otherwise} \end{cases} \quad (1)$$

where  $s = R + G + B$ .

These rules can be explained with the help of figure 2. For each pixel we calculated a set of secondary variables, i.e.  $R_N - G_N$  (Normalized Red value minus Normalized Green Value);  $G_N - B_N$  (Normalized Green value minus Normalized Blue value), etc.

Greek letters are representing threshold values. Each value was selected using the median value obtained in its case (figure 2). Letters  $R_{px}, A_{px}, G_{px}$  mean red pixel, amber pixel and green pixel respectively, which were sampled from the 60 images captured in diverse conditions. Each represented box is associated with one letter according to the color sampled.

For cluster 1 (red color) we show the results in figure 2a. Therefore, according to medians, we set up these parameters as:  $\alpha = 0.35$ ,  $\beta = 0.08$  and  $\gamma = 0.15$ . Just like in the aforementioned case, the adjusted parameters for cluster 2 (amber color) are:  $\delta = 0.5$ ,  $\omega = 0.15$  and  $\pi = 0.14$  (see figure 2b). Also, figure 2c shows the computed values for the third cluster (green color):  $\mu = -0.15$ ,  $\rho = 0.17$  and  $\epsilon = 0.25$ . Finally, the background traffic light pixel colors are not actually blacks ( $r, g, b = 0, 0, 0$ ) during day. Therefore, based in our group previous experience, we have set  $\varphi = 0.30$  for cluster 4.

After this process, we get four new isolated monochromatic images, which will be used in the following steps. Each image contains independently red pixels, amber pixels, green pixels and black pixels. Of course, other relationships between RGB components could be found, even other color spaces such as YUV can be used. However, we have found this clusterization mode in our process's first part successful.

### B. Traffic light detector - Algorithm

Traffic Light detection is accomplished by following a set of consecutive tasks:

1) *Filter Processing*: For each image, a number of candidate regions are detected alongside with a certain level of noise. Most noise can be easily removed using two morphological filters: Erosion and Dilation. Good results can

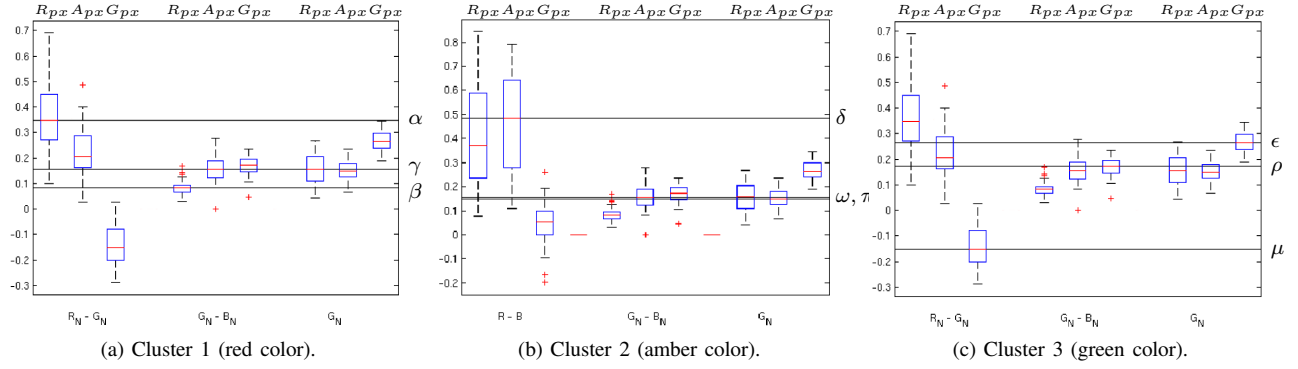


Fig. 2. Clusterization red, yellow and green color results.

0	255	0
255	255	255
0	255	0

Fig. 3. Dilation mask

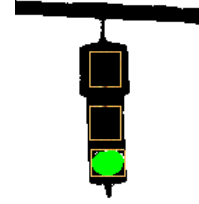


Fig. 4. Areas correctly detected by the function color pixel density

be achieved by an erosion followed by a dilation on the three color images<sup>1</sup>:  $\text{Img}' = ((\text{Img} - e) + d)$ , where  $e$  is the erosion and  $d$  is the dilation.

The erosion process is performed in order to preserve the detected shape. For each pixel the 8 neighbor pixels are considered: if the number of lighted pixels is bigger than a pre-established threshold value, the pixel value is set to the highest possible value (255). Otherwise, it will be set to the minimum possible value (0). Finally, the output is saved as a new image.

A dilation function is also performed. For each pixel the 8 neighbor pixels are considered: if the number of lighted pixels is bigger than another threshold value, the five pixels forming a cross centered on the tackled pixel (like the pattern shown in figure 3), are set to the maximum value (255).

2) *Labeling process*: This part is quite standard for every kind of visual detection methodology in literature. The three computed colors images, which contain red, amber and green information colors, are elaborated. All the connected areas in the image are detected using a flood fill algorithm ([7] shows some filling algorithm examples). Starting with the first detected pixel, search for connected pixel in all the 8 directions. Using the information of each label it becomes possible to build a bounding box which contains the whole area.

3) *Rules*: At this point, there are still many false detection points that must be filtered out. Some traffic light features, including also Italian traffic light regulations, are taken into account to discriminate between traffic lights and noisy points. Many noisy bounding box can be removed pursuant to visual features of traffic lights.

a) *First level. Bounding box features*: The bounding box size (width and height) is firstly considered. The Ital-

<sup>1</sup>Except in paragraph II-B3b, black pixel image is not processed in the rest of the algorithm in order to save time consumption.

ian traffic regulations establish the traffic lights diameters: 0.3m for red light and 0.2m for amber and green light, additionally, also the maximum and minimum height of the traffic light can be estimated. Therefore, using a perspective transformation, it is possible to compute, for a given distance, the size of the traffic lights in the image and the row in which they can be found as well.

Having fixed the minimum and maximum searching distance, and considering each distance in this range with a fixed coarseness, the higher and lower row in which traffic lights can be found and their maximum and minimum size are computed and stored. The idea is to find a range of possible diameters in the image, i.e. in pixels, for each row in the ROI: considering the stored information, these data can be computed separately for red, amber and green lights, and used to filter the detected bounding boxes.

Another obvious feature is that if the light is round, the bounding box will be square: height and width should be equals. This expression can therefore be considered, where  $\psi$  is just a threshold.

$$\frac{1}{\psi} \leq \frac{\text{height}}{\text{width}} \leq \psi \quad (3)$$

b) *Second level. Color pixel density*: An additional rule helping us with the detection consists on color density measurements, for both the traffic light dark frame and lighted colors. The process is based on the count the number of bright pixels that are in the bounding box. Assuming that  $N_{px}$  is the number of pixels and  $A$  is the area, we need to verify that this equation holds, where  $\chi$  is a threshold.

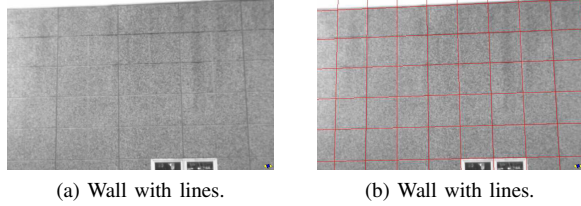


Fig. 5. Extrinsic parameters calibration.

$$\frac{N_{px}}{A} \leq \chi \quad (4)$$

When a light with a sufficient color pixel density is found, black pixel density is computed inside the area corresponding to the other lights: this area must be completely dark, since the other lights must be turned off. For instance, if a green light is detected, green pixel density within its bounding box is computed, then black pixel density is computed in an area corresponding to two bounding boxes above the detected one (see figure 4).

*c) Third level. Distance between car and traffic lights:* A lot of techniques to extract reliable information from input frames are present in computer vision. Thus, firstly we have calibrated the camera. The intrinsic parameters were calibrated using the classic Zhang method [8]. A reference grid drawn in a large wall was used to rectify the extrinsic parameters. Camera orientation were estimated adjusting pitch, roll and yaw values to make the lines match the drawn grid (see figure 5).

Stereo camera systems estimate distances from 2D images. However, if one world dimension (X, Y or Z) is fixed, the other two world dimensions could be computed from only one 2D image. In our work, we fix the height of the suspended traffic lights and we compute the (u, v). Finally, we can get the distance between vehicle and traffic light.

Also, in order to discriminate some noisy points, at this stage this simple rule is applied:

$$\rho \leq distance \leq \sigma \quad (5)$$

### C. Tracking Algorithm

Despite the simple tracking performed, this part provides robustness to both estimation and detection. Each traffic light candidate information (like its color, its position in the image, its centre of gravity, its distance, etc.) is stored for each frame. Each detected candidate can be matched to one of the previously detected traffic lights. It becomes possible to track the traffic lights, considering also their age, and the number of consecutive miss detections. Finally, if the age (number of frames in which the traffic light is found) exceeds a threshold, the bounding box is validated and shown on the image. On the other hand, if the lights are missed for a consecutive number of frames higher than a threshold, we remove them.

Additionally, an useful help to the tracking is developed. When a light is turned off and another one is turned on,

the two lights are supposed to lay approximatively in the same columns in the image. The distance between traffic light and vehicle should be similar too. Lighting-up order of the lights are also known: consequently, if an expected change is detected: the old light is removed and the new bounding box inherits its age.

## III. RESULTS

Algorithm explained in section II has supported the better trade-off between processing time and detection robustness. The average processing time for the whole algorithm is about 90 ms (on a Intel(R) Core(TM)2 Duo CPU @ 2.93GHz with 4GB RAM). The operative system used is Linux - Fedora Core 16 and the algorithm has been developed in C++. In the following paragraphs, we show our acquisition system and the detection rate is also presented together with some consideration about distance estimation.

### A. Setup acquisition system

For the presented research we have used a color camera with a  $752 \times 480$  pixels CMOS sensor (AVT guppy F-036C). Since the sensor is a single-chip with bayer filter, we need to perform a de-bayer transformation, in order to get color images. As the purpose of this work is to detect suspended traffic lights and to stop in front of red lights, the camera is positioned on the top of the roof in the rear part of the vehicle, and pitched up, as shown in figure 6.



Fig. 6. Camera position and orientation

Two big drawbacks of this mounting are that the camera is prone to dizziling and that it cannot be used for other purposes. Another important issue to be addressed in this kind of applications is the exposure time: if the time is too short, the traffic lights can be detected only when they are very close; however, if it is too long, saturation can cause problems in color clustering. Thus, the shutter must be correctly managed.

### B. Detection Rate

The results obtained during the tests are very promising. Except for a new LED traffic light. We suspect its green light is too close to blue to be well detected by using the calculated threshold values. Unfortunately, the number of false positive detections is quite high. Out of 27000 frames, both with and without traffic lights, we correctly achieved

14000 detections. In a frame we could find one or two suspended traffic lights. Traffic lights can be detected up to 80 meters, but the maximum guarantee detection distance is 50 meters, for very poor visibility conditions. We have detected 1500 false positives: in most of the cases this is due to little and far amber clusters which are present for a little number of frames; only in a couple of cases the false positives are persistent.

### C. Distance Estimation

This work is focused on suspended traffic lights detection and distance estimation: we have considered suspended traffic lights located at a fixed height. The weakness of this technique is when the height of the traffic lights are not equal to the one considered, as often happens in the urban area analyzed. According to Italian road laws, the height of traffic lights could change in a determinate range. Therefore, we have simulated the variation of distance according to this height range: error is directly proportional to the distance and it reaches acceptable value in the last 50 meters.

Another method for distance estimation was investigated; this method is based on the assumption that the traffic lights have a fixed diameter and that they are completely and precisely detected by the algorithm: interesting, although not sufficiently precise, results were reached.

An experiment were carried out for the distance estimation. Starting at distance of 70 m (measured from the front of the camera to the suspended traffic lights), we placed the car at 7 different distances, getting 10 m closer each time. At every position we captured images of a whole traffic light cycle. These 7 distances were measured by using marks on the road. The distances were computed and the results are shown in figure 7. Colors in the graphics refer to detected traffic light color.

It can be observed that the distance detection was not very accurate at a distance of 70 meters. Results improve for the rest of distances. It is interesting to notice that for the first traffic cycle captured, two green traffic light distances were detected (around frame  $1.5 \cdot 10^3$ ), we can see two different measurements: this is because we have two suspended traffic lights. When green lights are switching-on, the bounding boxes detected are not equal for this case. We suspect that, for instance, this could be resolved by using distance data detected earlier. This happens in many parts of our graphics, as we can see, especially for green and red color in some parts. Another matter that we need to resolve occurs when the car changes from static to motion and vice versa. This produces a change in the pitch of the camera. The effect is shown as a small peak in several parts of the graphic. Maybe if we apply a pitch compensation, it could be enough to mitigate this difficulty. Also, the detection is missed around 15 m.

We have observed in the sequence that the light was hidden by a tree limb in that moment. We have found two sets of points detected, which have not been filtered out: one in

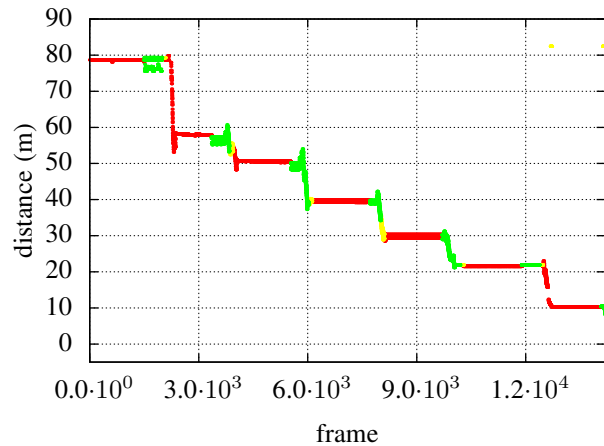


Fig. 7. Detection training during the evening.

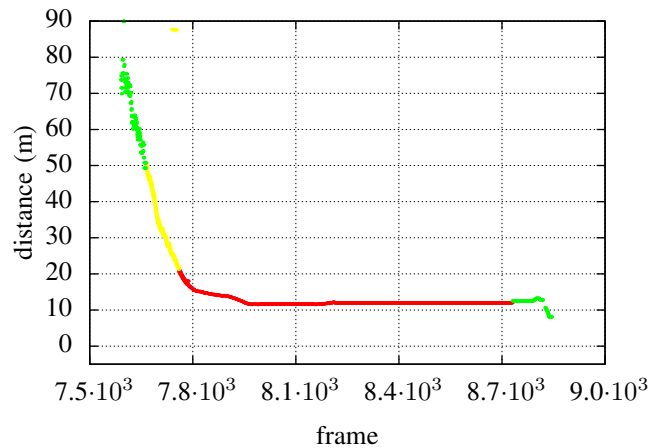


Fig. 8. Distance Estimation when the vehicle finds a suspended traffic light changing its color.

$1.3 \cdot 10^4$  and the other at the end of the graphics. Both of them represent the amber light. Some judgement based on distance could be useful to filtrate them. Finally, when the distance is around 8 m, the traffic lights disappear from our vision field.

In figure 8 we represent the results of another experiment. It includes the approaching to a traffic light during a complete cycle. In this test we drove the car normally approaching a traffic light on green, then it turned amber and red, we stopped at approximately 10 m distance, then it turned green and we moved on. In that picture it can be observed that the distance accuracy becomes better and better as we get closer to the traffic light.

### D. Qualitative results

Figure 9 shows the procedure on a frame. In image 9a the red light is quite distinguishable between other colors. Then the colors after the clustering are shown in image 9b. This image clearly shows some detectable color features, like the

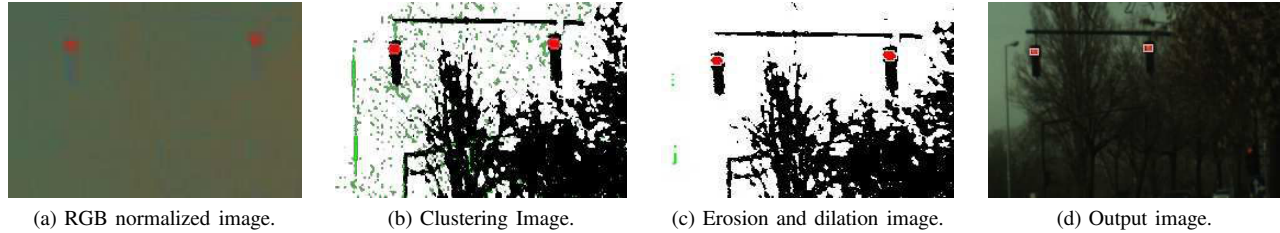


Fig. 9. Red detection and distance estimation procedure on a frame

black area belonging to suspended traffic lights or the tube where they are hanging. Despite time consumption is raised, the image 9c manifests the importance of morphological operations in the algorithm. After performing the filtering according to the color features and the tracking stage, the final results can be seen in image 9d.

The algorithm was tested in several brightness conditions, during day and night and other climatic conditions. The method works pretty well during the evening, when the sun is less bright. When the sun is present, images need to compensate the brightness challenge. An adaptive shutter parameter according to the color level on the top of our image will be developed in future works. Additionally, our method is weak during the night. Streetlights introduce a lot of false positive points in our system, particularly for amber color. Indeed the brightness of the lights increases the correct detection during the night. The shutter parameters have been changed during several acquisitions. We strongly believe that if we use a low shutter parameter, the light detected will be more easy to find.

Finally, this method seems to detected suspended traffic lights located up to 80 meter distance. Also, the distance is correctly estimated when the car is located up to 60 meters from the traffic lights distance. These two parameters have been interesting to filter out noisy points.

#### IV. CONCLUSIONS AND FUTURE DEVELOPMENTS

A technique to detect suspended traffic lights and estimate their distance, working in real time, is presented. The work focuses not only on detection, but also on distance estimation. According to this feature, it becomes less difficult to cope with very complex scenarios, in which several traffic lights are present (figure 10 shows an example of complex situations for traffic light detection). We believe that our distance estimation method could tackle such situation, such as by accepting those traffic lights which are closer to the vehicle.

Particular attention was paid to the processing time: erosion and dilation functions consume approximately 40 % of total execution time. However, better results can be reached using both these functions. Computational times must be reduced, for instance, by parallelizing both functions.

This work is based on the assumption of a traffic light with a black shape and only one lamp turned on while the



Fig. 10. Via Vittor Pisani, Milano

rest of the lamps are turned off. Of course, in this case, we need to include other rules. Indeed, during tests the shutter was manually set, but for real application an auto adjustable shutter must be developed.

Finally we have some ideas based on ego motion computation as an attempt to compensate the pitch effect according to subsection III-C, and to strengthen the tracking stage.

#### REFERENCES

- [1] C. Yu, C. Huang, and Y. Lang, "Traffic light detection during day and night conditions by a camera," in *Signal Processing (ICSP), 2010 IEEE 10th International Conference on*, oct. 2010, pp. 821–824.
- [2] Y. Shen, U. Ozguner, K. Redmill, and J. Liu, "A robust video based traffic light detection algorithm for intelligent vehicles," in *Intelligent Vehicles Symposium, 2009 IEEE*, june 2009, pp. 521–526.
- [3] N. Fairfield and C. Urmson, "Traffic light mapping and detection," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, may 2011, pp. 5421–5426.
- [4] R. de Charette and F. Nashashibi, "Traffic light recognition using image processing compared to learning processes," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, oct. 2009, pp. 333–338.
- [5] M. Omachi and S. Omachi, "Detection of traffic light using structural information," in *Signal Processing (ICSP), 2010 IEEE 10th International Conference on*, oct. 2010, pp. 809–812.
- [6] —, "Traffic light detection with color and edge information," in *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, aug. 2009, pp. 284–287.
- [7] A. Glassner, "Fill 'er up! [graphics filling algorithms]," *Computer Graphics and Applications, IEEE*, vol. 21, no. 1, pp. 78–85, jan/feb 2001.
- [8] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330–1334, nov 2000.